3D VISION

Structure from Motion:
Optical Flow

Systems Engineering and Automation

Miguel Hernández University

Table of Contents

Introduction

- Input data: sequence of images taken at different time instants

- What information can be obtained from changes in sequence of images taken at different time instants?

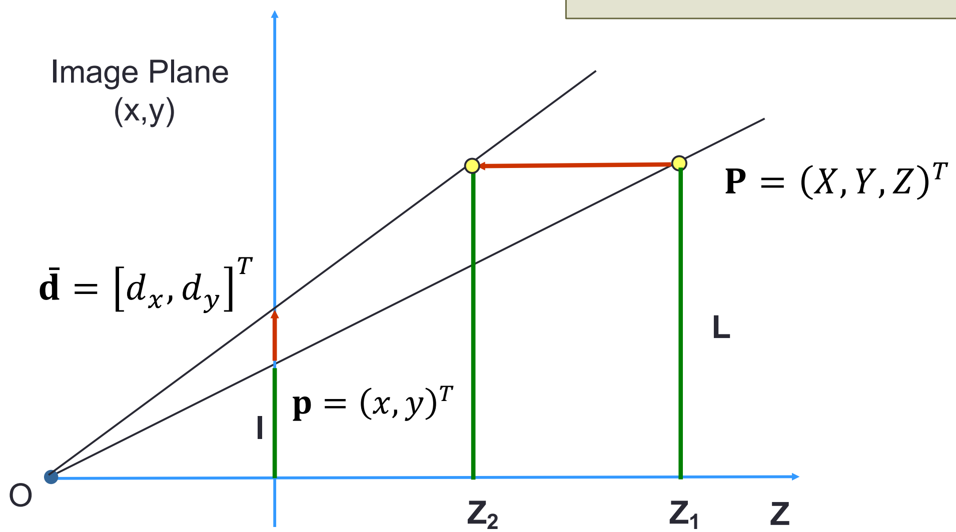
- Assuming constant illumination, changes in images can be caused by relative motion between camera and scene:
  - Moving camera in a static scene
  - Moving objects from the scene in front of a stationary camera
  - Motion of camera and objects in the scene relative to an extern reference

Importance of Visual Motion

- Reasons to introduce time variable to image processings:
  - Apparent motion of objects in the image → understand the structure of objects and their movement in 3D space
  - Biological visual systems use visual motion to infer 3D space properties with a small a priori knowledge (time to impact)

- Example :

$$\bar{\mathbf{d}} = \left[d_x, d_y\right]^T = f(\bar{\mathbf{v}} \cdot \delta t) = f'(Z_2 - Z_1)$$



Motion Analysis Problem

- **Correspondence:** What elements between different images in the sequence correspond to the same 3D primitive? (Motion Field)
  - Methods to solve the correspondence problem:
    - **Differential methods**: dense measurements, calculated for every point in the image. High image sampling sequence. Optical Flow
    - **Feature matching methods**: sparse measures, calculated only for a limited set of points in the image. Kalman filter. They use optical flow as an initial estimate.

- **Reconstruction:** involves getting the velocity maps of the objects in the 3D scene

Motion Analysis Problem

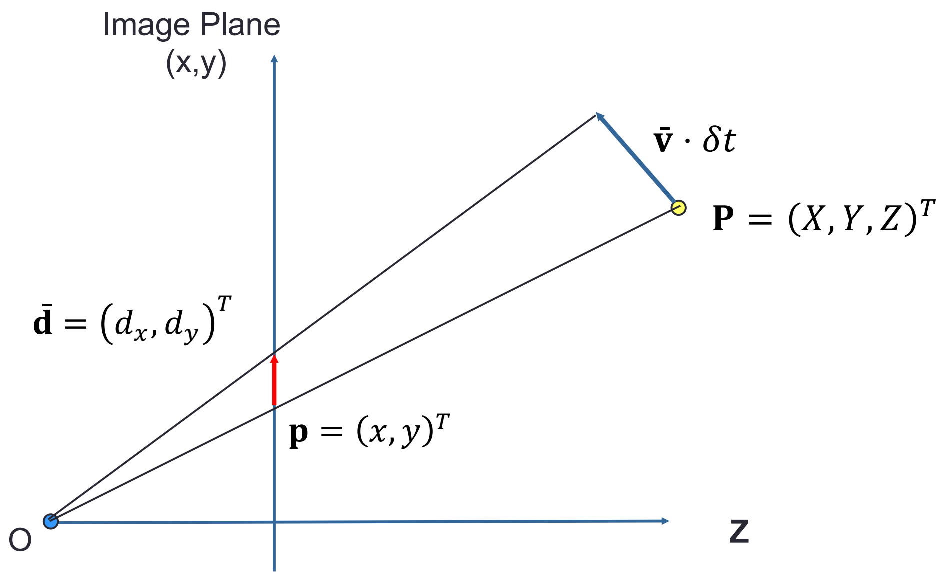
- Restriction : there is a single relative rigid motion between the camera and the observed scene, and the lighting conditions do not change.
  - Example:
    - YES: views of a building by a moving observer
    - NO: football sequences, traffic,..
    - NO: deformable objects

- There are several moving objects. The problem of Segmentation arises.
  - What are the regions of the image that correspond to different moving objects?
  - Approaches:
    - Segment → Correspondence
    - Correspondence → Segment.

Table of Contents

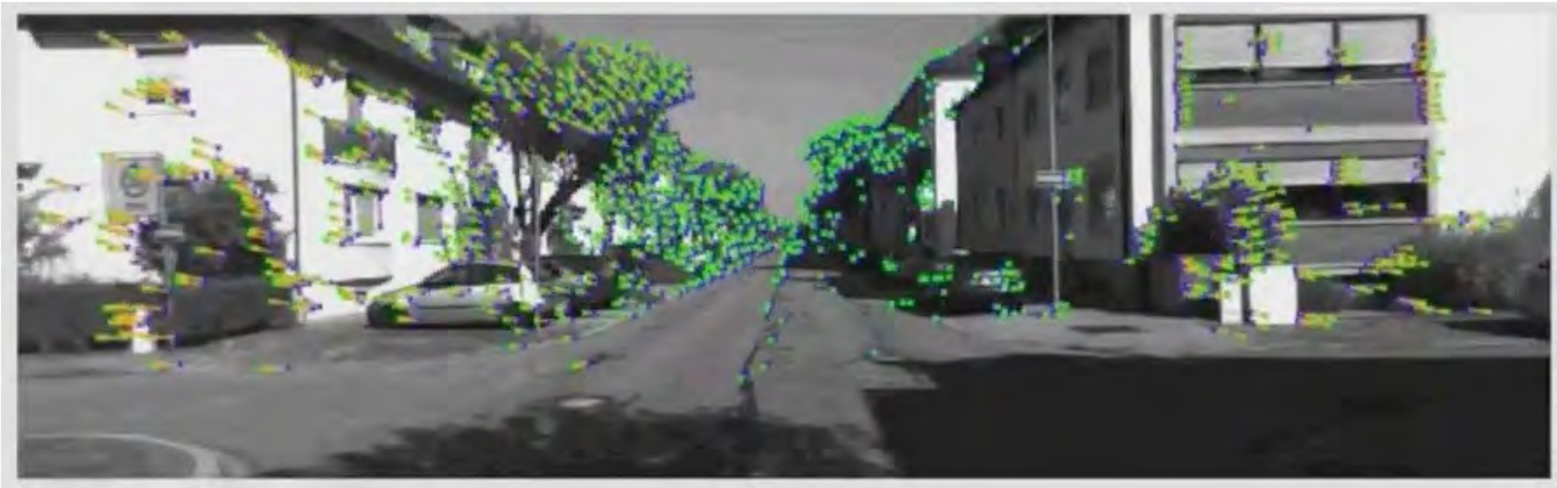
Motion Field

- Motion Field:
  - Definition : It is the 2D velocity vector field of the points in the image, induced by the relative movement between the camera and the observed scene.
  - It can be seen as the projection of the 3D velocity field on the plane of the image.



Motion Field

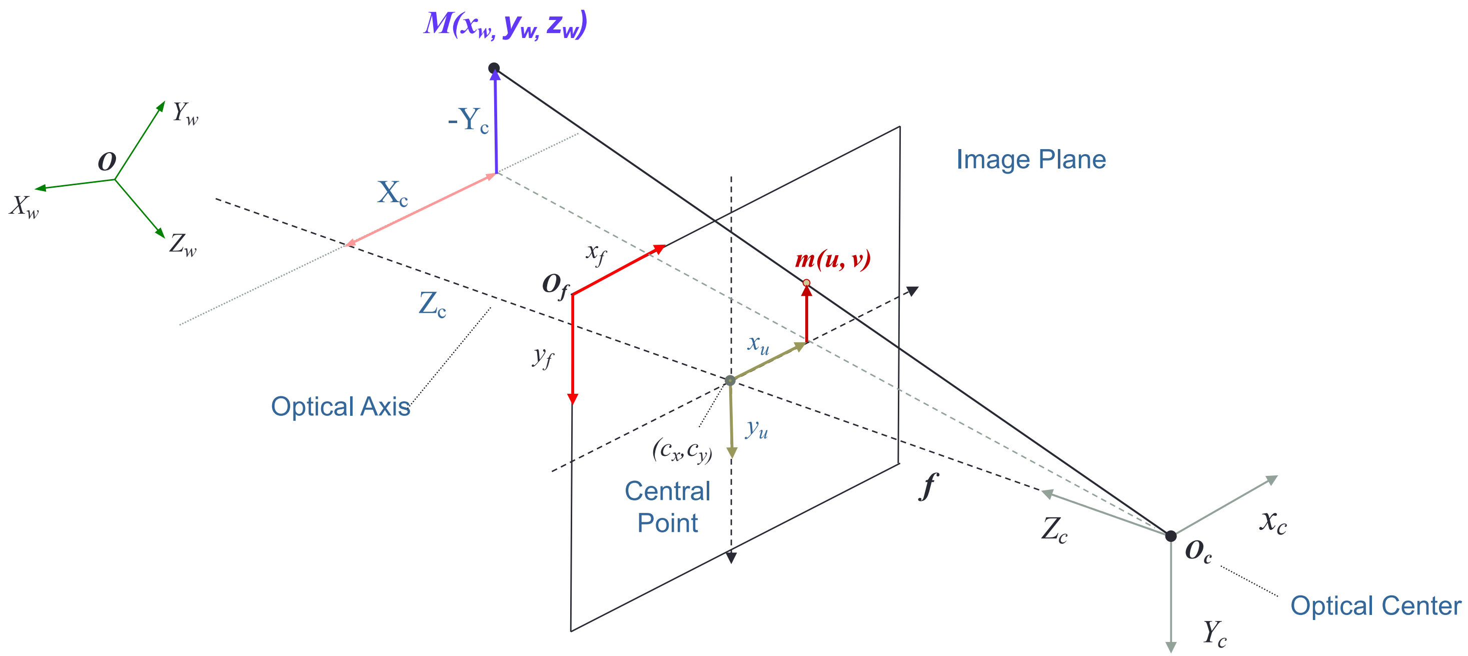
- Example:



(Penn Engineering Univ. Pennsylvania)

Motion Field

- The image coordinates will be considered with respect to a Reference System centered on the optical center (central point).
- We will consider known the intrinsic parameters



Motion Field

- Projection equations without considering distortion:
  - Normalized coordinates:

$$\tilde{m} \cong A\ [I \quad 0]\ \tilde{M}_c \quad \Rightarrow \quad \tilde{p} = A^{-1}\tilde{m} \cong\ [I \quad 0]\ \tilde{M}_c$$

$$\begin{bmatrix} n\,x \\ n\,y \\ n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \qquad x = \frac{X_c}{Z_c} \qquad y = \frac{Y_c}{Z_c}$$

$$\Downarrow$$

$$\overline{p}(t) = \frac{\overline{P}(t)}{Z_c} \qquad \overline{p}(t) = \frac{\overline{P}(t)}{\overline{P}(t)\cdot\overline{k}}$$

$$\overline{p} = [x\,,\ y,\ 1]^{\,T}$$

$$\overline{P} = [X_c,\ Y_c,\ Z_c]^{\,T} \qquad \overline{k} = [0,\ 0,\ 1]^{\,T}$$

Motion Field

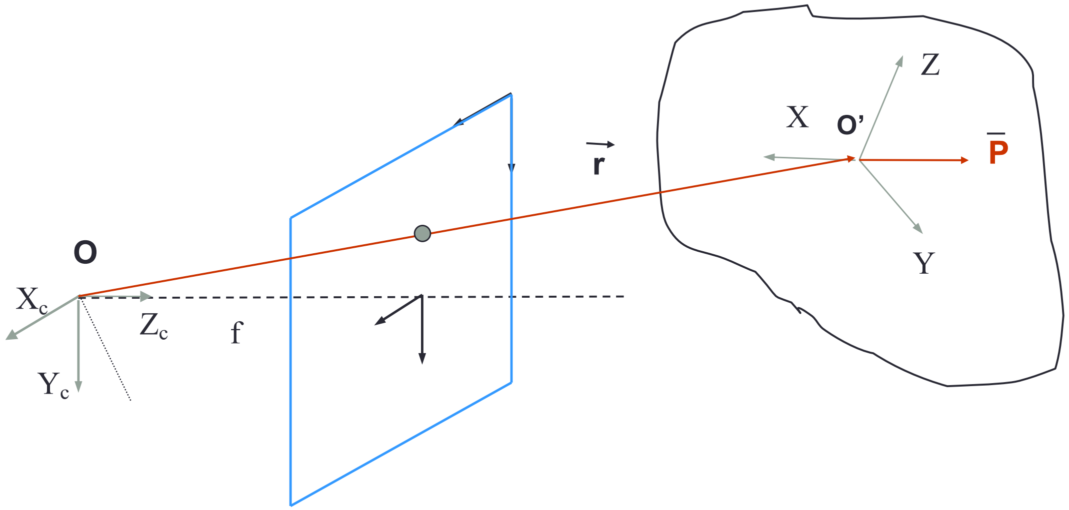
- The relative movement between the point **P** and the camera is broken down into a translation and a rotation:
  - Kinematics of a rigid body:

$$\bar{V} = -\frac{d\overline{OP}(t)}{dt} = -\bar{T} - \bar{\omega} \times \overline{O'P}$$

$$V_x = -T_x - \omega_y Z + \omega_z Y$$
$$V_y = -T_y - \omega_z X + \omega_x Z$$
$$V_z = -T_z - \omega_x Y + \omega_y X$$

$\bar{T}$: translational velocity

$\bar{\omega}$: rotacional velocity



Motion Field

- Motion field: derivation with respect to time:

$$\bar{v}(t) = \frac{d\bar{p}}{dt} = \frac{d}{dt}\left(\frac{\bar{P}}{Z}\right) = \frac{Z\bar{V} - V_z\bar{P}}{Z^2} \quad \Rightarrow \quad \bar{v}(t) = \frac{d\bar{p}}{dt} = \frac{\left(\bar{P}\cdot\bar{k}\right)\bar{V} - \left(\bar{V}\cdot\bar{k}\right)\bar{P}}{\left(\bar{P}\cdot\bar{k}\right)^2}$$

more compactly

$$\boxed{\bar{v}(t) = \frac{d\bar{p}}{dt} = \frac{1}{Z(t)^2}\left(\bar{P}(t)\times\bar{V}(t)\right)\times\bar{k}}$$

$$\left(\bar{a}\times\bar{b}\right)\times\bar{c} = (\bar{a}\cdot\bar{c})\ \bar{b} - \left(\bar{b}\cdot\bar{c}\right)\ \bar{a}$$

Motion Field

- Taking components :

$$v_x = \frac{T_z x - T_x}{Z} - \omega_y + \omega_z y + \omega_x xy - \omega_y x^2$$
$$v_y = \frac{T_z y - T_y}{Z} + \omega_x - \omega_z x - \omega_y xy + \omega_x y^2$$

- Two components: that depend on translation and rotation respectively

$$v_x^T = \frac{T_z x - T_x}{Z}$$

$$v_x^\omega = -\omega_y f + \omega_z y + \omega_x xy - \omega_y x^2$$

$$v_y^T = \frac{T_z y - T_y}{Z}$$

$$v_y^\omega = \omega_x f - \omega_z x - \omega_y xy + \omega_x x^2$$

Motion Field

- The component of the motion field in the direction of the optical axis is zero

$$\bar{v} = \left[v_x,\ v_y\right]^T \quad \Rightarrow \quad \bar{v} = \left[v_x,\ v_y,\ 0\right]^T \qquad \bar{p} = [x\ ,\ y,\ 1]^T$$

- From the previous equations we get that $\bar{\omega}$ (angular velocity) is decoupled from Z (Depth)
  - The part of the motion field that depends on angular velocity has no implicit information about depth

$$v_x^T = \frac{T_z x - T_x}{Z} \qquad v_y^T = \frac{T_z y - T_y}{Z}$$

Motion Field

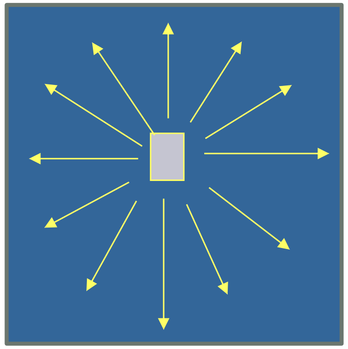
- CASES : Pure Translation ($\overline{\omega} = 0$)
  - T_z nonzero.

$$v_x^T = \frac{T_z x - T_x}{Z} \qquad v_y^T = \frac{T_z y - T_y}{Z}$$

- Calling

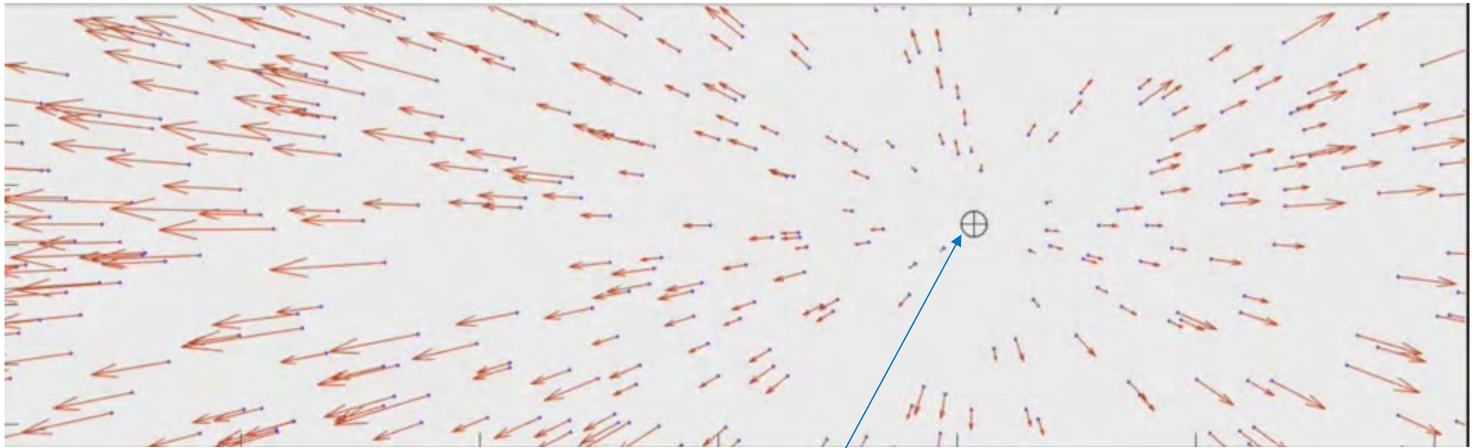
$$\begin{cases} \overline{p}_0 = [x_0, y_0]^T & \text{Vanishing Point} \\ x_0 = \dfrac{T_x}{T_z} \qquad y_0 = \dfrac{T_y}{T_z} \end{cases} \Rightarrow \quad v_x^T = (x - x_0)\frac{T_z}{Z} \qquad v_y^T = (y - y_0)\frac{T_z}{Z}$$

- **The motion field for a pure translation is RADIAL. Vectors with a radial disposition from a vanishing point $\overline{\boldsymbol{p}}_0$**
  - If $T_z < 0 \Rightarrow \overline{\boldsymbol{p}}_0$ is an expansion focus
  - If $T_z > 0 \Rightarrow \overline{\boldsymbol{p}}_0$ is a contraction focus
- **$\overline{\boldsymbol{p}}_0$ is the intersection of the ray parallel to the translation vector with the plane of the image**



Motion Field

- Pure Translation ($\overline{\omega} = 0$)
  - The length of $\overline{v} = \overline{v}(p)$ proportional to the distance between $\overline{p}$ and $\overline{p}_0$ (vanishing point), and inversely proportional to the depth of the point $\overline{P}$ in 3D space



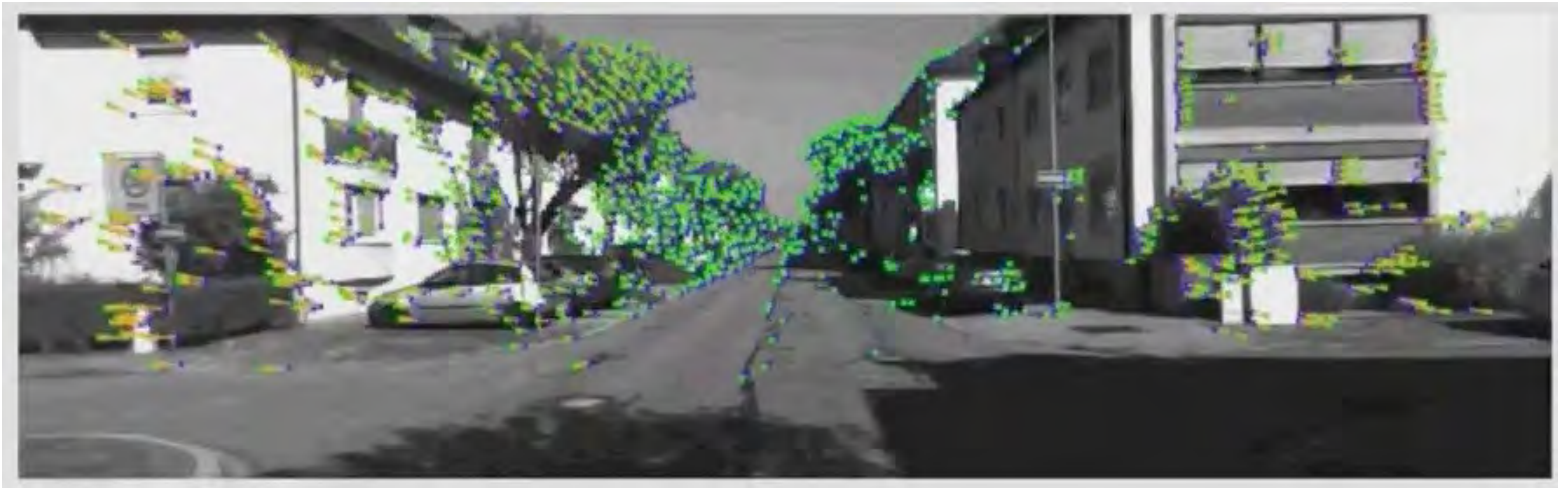
$$v_x^T = (x - x_0)\frac{T_z}{Z}$$

$$v_y^T = (y - y_0)\frac{T_z}{Z}$$

$$\begin{cases} \overline{p}_0 = [x_0, y_0]^T \\ x_0 = \dfrac{T_x}{T_z} \qquad y_0 = \dfrac{T_y}{T_z} \end{cases}$$

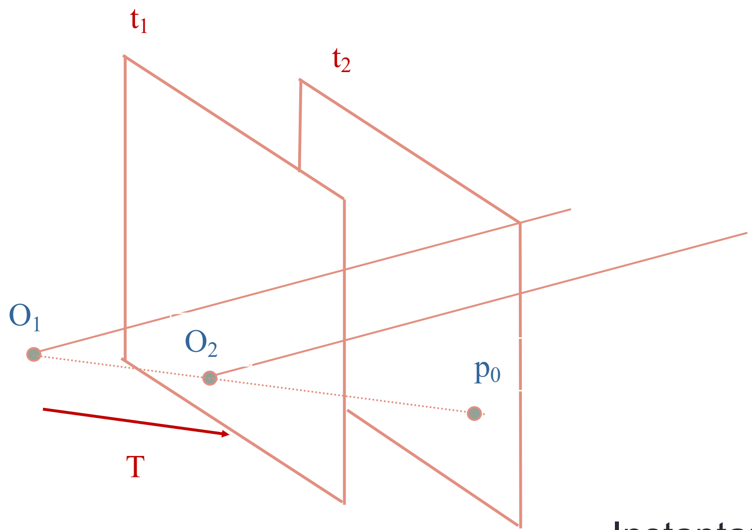
Motion Field

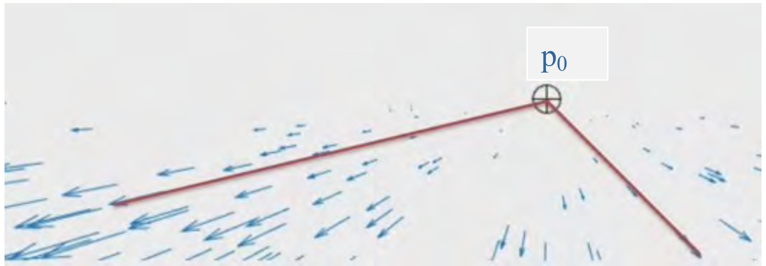
- Example:



(Penn Engineering Univ. Pennsylvania)

- Instant Epipole
  - It is possible to locate $\overline{p}_0$ without an apriori knowledge of the intrinsic parameters of the camera.
    - If we move the camera O_1 to O_2 with a pure translation, $\overline{p}_0$ will be the point in the image whose motion field is null.





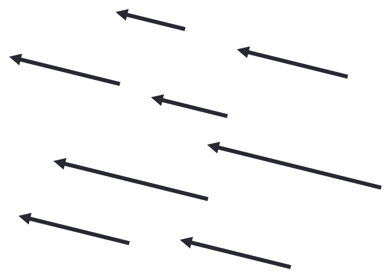
Instantaneous Epipole

$$\begin{cases} \overline{p}_0 = [x_0, y_0]^T \\ x_0 = \dfrac{T_x}{T_z} \qquad y_0 = \dfrac{T_y}{T_z} \end{cases}$$

Motion Field

- Pure Translation ($\overline{\omega} = 0$) $$v_x^T = \frac{T_z x - T_x}{Z} \qquad v_y^T = \frac{T_z y - T_y}{Z}$$
  - If $T_z = 0$, all vectors in the motion field are parallel and their lengths are inversely proportional to the depth.
  - The Instant Epipole $\overline{p}_0$ is at infinity

$$v_x^T = -\frac{T_x}{Z} \qquad v_y^T = -\frac{T_y}{Z}$$



Motion Field

- Pure Translation ($\overline{\omega} = 0$)
  - **Time to Collision** from the optical flow (motion field)
  - From the optical flow and the instant epipole $\overline{p}_0$ we can calculate the collision time (used by birds and insects to determine the flight velocity)

$$v_x^T = (x - x_0)\frac{T_z}{Z}$$

$$v_y^T = (y - y_0)\frac{T_z}{Z}$$

$$\overline{v}^T = \frac{T_z}{Z}\begin{bmatrix}(x - x_0)\\(y - y_0)\end{bmatrix}$$

$$TTC(seconds) = \frac{Z}{T_z} = \frac{\left\|\overline{p} - \overline{p}_0\right\|}{\left\|\overline{v}^T\right\|}$$

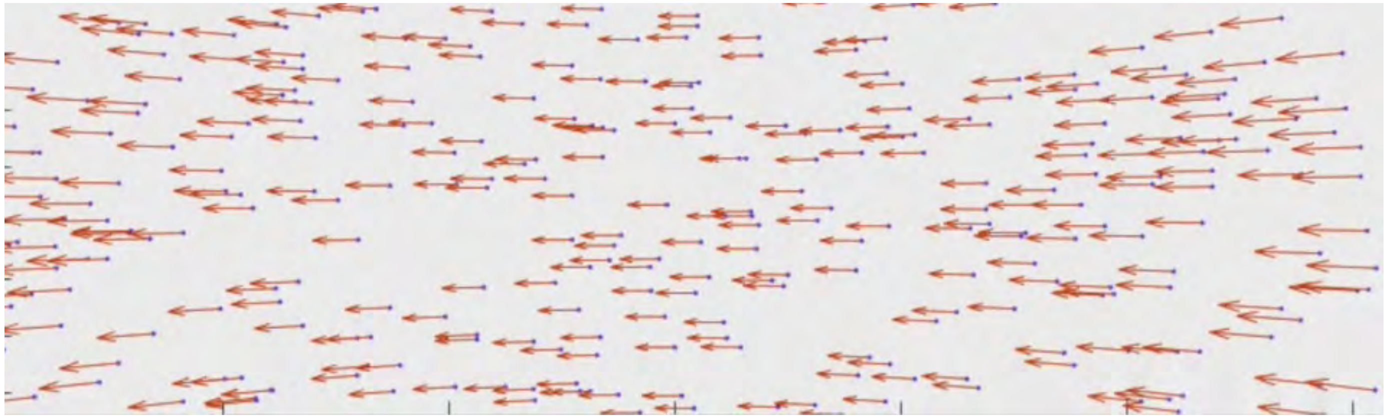
$$\overline{p} = [x, y]^T$$

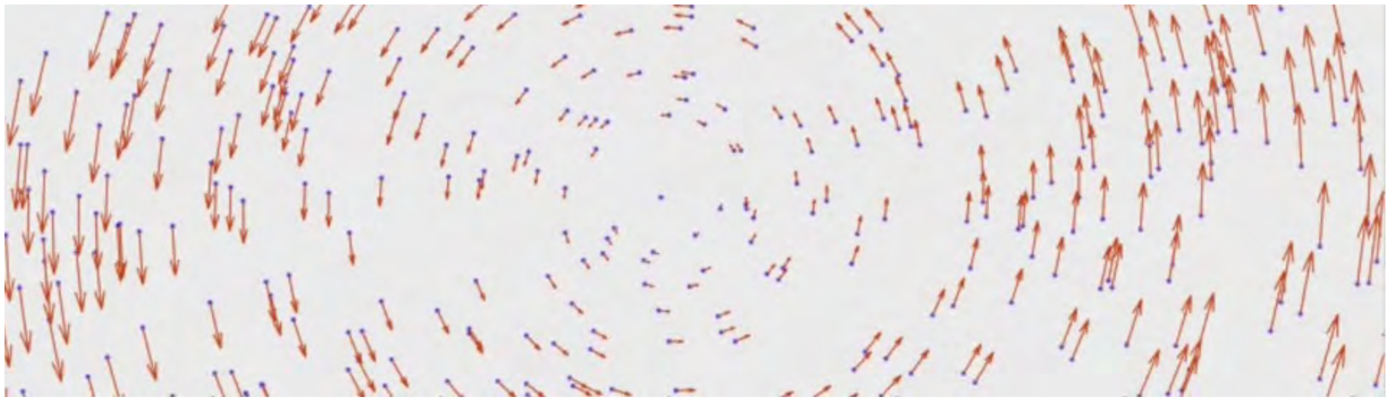
$$\overline{p}_0 = [x_0, y_0]^T$$

For points at the same distance from the vanishing point, the collision time is inversely proportional to the length of the optical flow

Motion Field

- Pure rotation ($\bar{T}=0$)

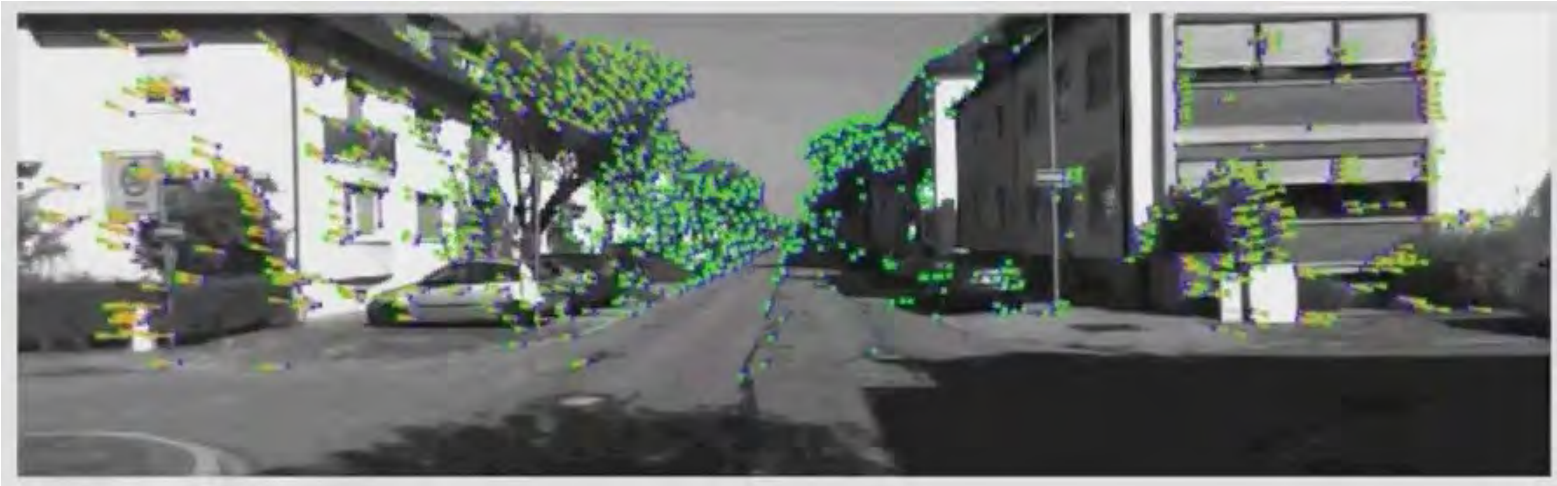




$$v_x = -\omega_y + \omega_z y + \omega_x xy - \omega_y x^2$$
$$v_y = \omega_x - \omega_z x - \omega_y xy + \omega_x y^2$$

Motion Field

- Example:



(Penn Engineering Univ. Pennsylvania)

Motion Field

- Movement of a Plane
  - We look at a planar surface

$$\bar{n}^T \cdot \bar{P} = d \quad \bar{n}^T = [n_x, n_y, n_z] \Rightarrow \bar{p} = \frac{\bar{P}}{Z} \Rightarrow (n_x x + n_y y + n_z) \cdot Z = d$$

  - $\bar{n} \rightarrow$ Normal to plane
  - $d \rightarrow$ distance from the plane to the projection center.
  - The flat surface has a movement $(\bar{T}, \bar{\omega})$
  - Clearing Z and replacing in the general equation of the motion field:

$$v_x = \frac{T_z x - T_x}{Z} - \omega_y + \omega_z y + \omega_x xy - \omega_y x^2$$
$$v_y = \frac{T_z y - T_y}{Z} + \omega_x - \omega_z x - \omega_y xy + \omega_x y^2$$

$$v_x = \frac{1}{d}(a_1 x^2 + a_2 xy + a_3 x + a_4 y + a_5)$$
$$v_y = \frac{1}{d}(a_1 xy + a_2 y^2 + a_6 y + a_7 x + a_8)$$

$$a_1 = -d\ \omega_y + T_z n_x \quad a_2 = d\ \omega_x + T_z n_y$$
$$a_3 = T_z n_z - T_x n_x \quad a_4 = d\ \omega_z - T_x n_y$$
$$a_5 = -d\ \omega_y - T_x n_z \quad a_6 = T_z n_z - T_y n_y$$
$$a_7 = -d\ \omega_z - T_y n_x \quad a_8 = d\ \omega_x - T_y n_z$$

Motion Field

- Movement of a Plane

- Conclusions :
  - The motion field of a planar surface is a quadratic polynomial in coordinates of the image
  - Due to the spatial symmetry of polynomial coefficients, the same velocity field can be produced by different planar surfaces under different movements in space
  - To recover movement in the 3D space, we cannot relay on coplanar points.

Motion Field

- Aperture Problem:
  - The image is a limited projection of the scene. Doesn't always provide enough information about the global movement of an object

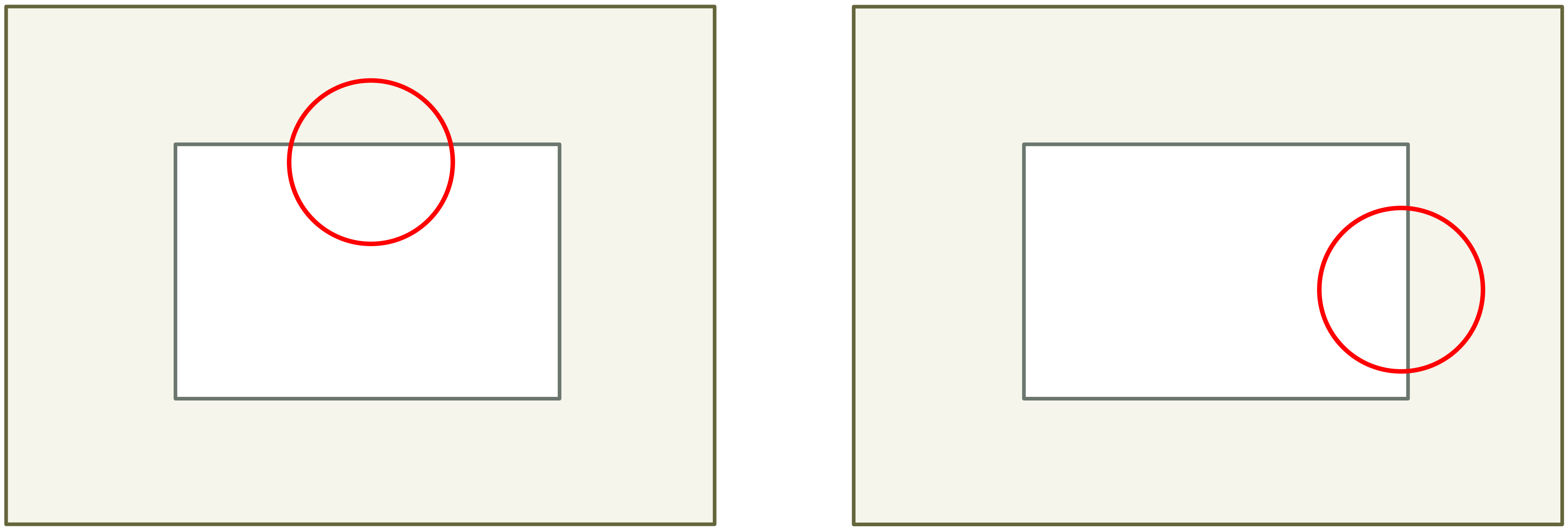


Table of Contents

Optical Flow

- We want to estimate the motion field of image sequences, based on spatial and temporal variations of the Image intensity

- Image Intensity Continuity Equation
  - E (Apparent image intensity) remains constant dE/dt = 0

$$E(x(t), y(t), t) \Rightarrow \frac{dE(x(t), y(t), t)}{dt} = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0$$

$$\boxed{(\nabla E)^T \cdot \overline{v} + E_t = 0 \qquad where \;\; E_t = \frac{dE}{dt}}$$

  - Aperture problem: “The motion field component in the orthogonal direction to the spatial gradient of the image, is not constrained by the image intensity continuity equation”

Optical Flow

- Optical Flow is a vector $\bar{\boldsymbol{v}}$ constraint by the previous equation (image intensity continuity equation), and is defined as the apparent motion of the image intensity pattern.

- It formally expresses that a point on a moving object will be projected at different points in the image at different times, but the irradiated intensity is the same at both projected points.

$$\nabla_x E \cdot v_x + \nabla_y E \cdot v_y + \nabla_t E = 0$$

$$I_1(x, y) = I_2(x + d_x, y + d_y)$$

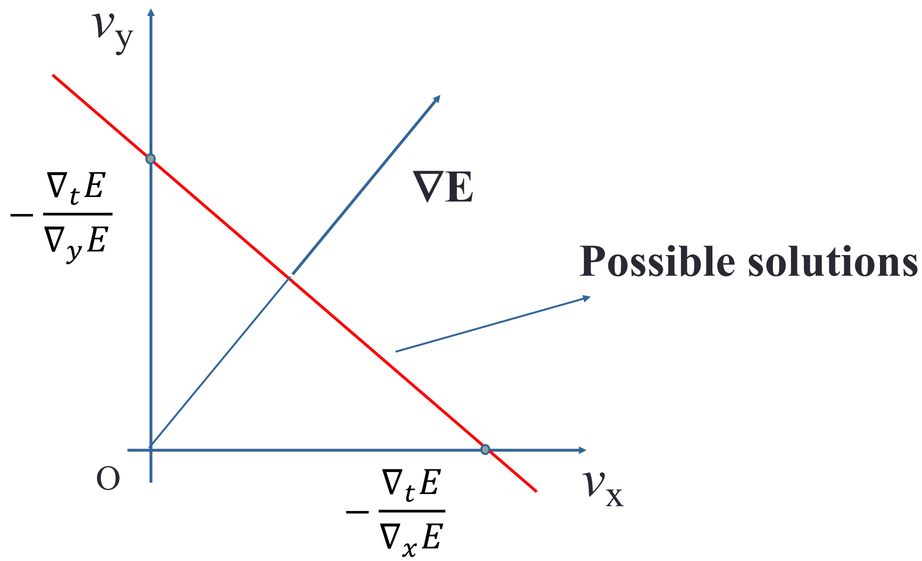
$$(d_x, d_y) = \bar{\boldsymbol{v}}\,(x, y) \cdot \delta t$$

Optical Flow

- Solution of the Image Intensity Continuity equation:

$$\nabla_x E \cdot v_x + \nabla_y E \cdot v_y + \nabla_t E = 0$$

  - Line in the velocity space
  - Ambiguous solution due to the aperture problem
  - Restrictions need to be imposed



Optical Flow

- Optical Flow and Motion Field
  - Optical flow is a good approximation of the motion field
  - It can be calculated from image sequences.
  - The approach is good with the following assumptions:
    - Lambertian surfaces (reflects the same brightness in all directions)
    - Point light source at infinity (very far)
    - No photometric distortion
  - The error of this approach is:
    - Small at points with a high spatial gradient
    - Zero only for translational motion or for any rigid motion such that the lighting direction is parallel to the angular velocity

Optical Flow

- The optical flow corresponds (approximately) to the projection of the motion field over the gradient vector of the image.

$$\nabla_x E \cdot v_x + \nabla_y E \cdot v_y + \nabla_t E = 0$$

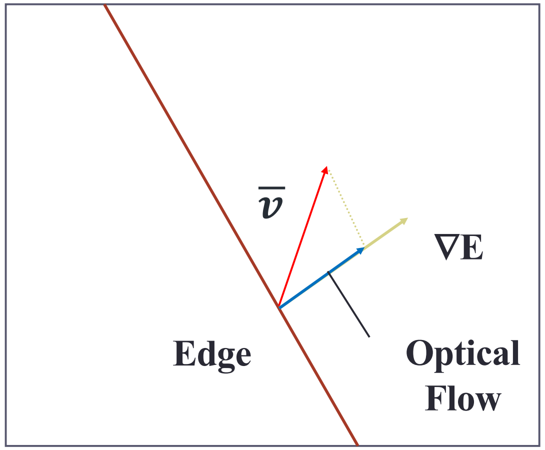


Table of Contents

Estimation of the Motion Field

- Techniques :
  - Differential techniques: based on temporal and spatial variations in image intensity for all the image points ⇒ Optical Flow
    - Lucas-Kanade
    - Horn-Schunck
    - Farneback
  - Correspondence techniques: determine the disparity between features in the image sequence. They can use two or more images (Kalman filter). They use optical flow as an initial estimate of the disparity.

Estimation of the Motion Field

- Differential techniques: (Lucas-Kanade)
  - Estimate optical flow by least squares (*restriction of local uniformity*)
  - Reasons:
    - Non-iterative, less prone than iteratives methods to possible discontinuities
    - They do not use derivatives in order greater than one, less sensitive to noise
  - Assumptions:
    - The image intensity continuity equation is a good approximation of the normal component of the field of motion.
    - The motion field will be approximated by a field of constant vectors within a small region Q of the image.

"An iterative image registration technique with an application to stereo vision" B. D. Lucas and T. Kanade, Proceedings of Imaging Understanding Workshop, (1981)

Differential Techniques (Lucas-Kanade)

- Algorithm for optical flow calculation (locally constant)
  - Typical neighborhood region 5x5(**Q**), minimize this functional:

$$\min_{\overline{v}} \psi(\overline{v}) \; ; \qquad \psi(\overline{v}) = \sum_{p_i \in Q} [(\nabla E)^T \cdot \overline{\boldsymbol{v}} + E_t]^2$$

$$A_{N^2 \times 2} = \begin{bmatrix} \nabla E(\overline{p}_1) \\ \nabla E(\overline{p}_2) \\ . \\ . \\ \nabla E(\overline{p}_{NxN}) \end{bmatrix} \; ; \; b_{N^2} = \begin{bmatrix} -E_t(\overline{p}_1) \\ -E_t(\overline{p}_2) \\ . \\ . \\ -E_t(\overline{p}_{NxN}) \end{bmatrix} \qquad \boxed{A\ \overline{v} = b}$$

N: tamaño de la ventana de integración Q

$$A^T A = \begin{bmatrix} \sum (\nabla_x E)^2 & \sum \nabla_x E \cdot \nabla_y E \\ \sum \nabla_x E \cdot \nabla_y E & \sum (\nabla_y E)^2 \end{bmatrix}$$

  - Using least squares

$$\begin{aligned} Sol: &\quad \overline{v} = (A^T A)^{-1} A^T b \\ \overline{v} \ = &\quad flujo\ optico\ en\ el\ centro\ de\ la\ región\ Q \end{aligned}$$

$$A^T b = \begin{bmatrix} -\sum \nabla_x E \cdot E_t \\ -\sum \nabla_y E \cdot E_t \end{bmatrix}$$

  - It is evaluated for all points in the image

"An iterative image registration technique with an application to stereo vision" B. D. Lucas and T. Kanade, Proceedings of Imaging Understanding Workshop, (1981)

Differential Techniques (Lucas-Kanade)

- Improving the algorithm:
  - An error is made when considering that the optical flow in the region Q is constant and equal to that of the center point, the error increases with the distance of the pixel to the center point. We introduce a weight matrix $W(N^2 \times N^2)$ symmetrical

$$\overline{v}_w = (A^T W^2 A)^{-1} A^T W^2 b$$

- Points in the image where the algorithm fails
  - When the matrix (2x2) $A^T A$ is singular in the region Q
  - Spatial gradients in Q null or parallel (colinear).
  - In this case the aperture problem has no solution.

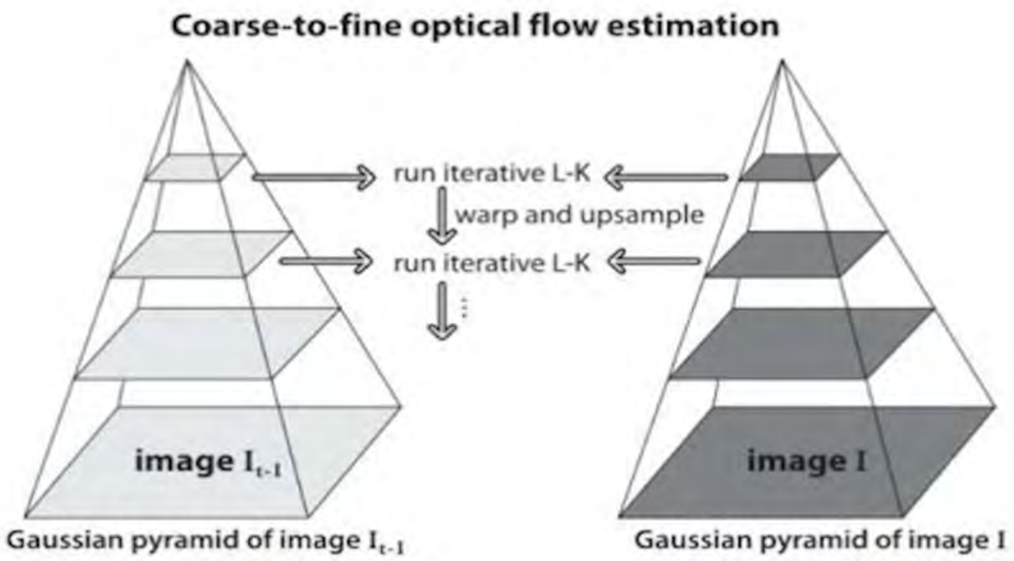
$$A^T A = \begin{bmatrix} \sum (\nabla_x E)^2 & \sum \nabla_x E \cdot \nabla_y E \\ \sum \nabla_x E \cdot \nabla_y E & \sum (\nabla_y E)^2 \end{bmatrix}$$

- Solution: calculate optical flow only at points where the eigen values of $A^T A$ are significant (Shi y Tomasi)

"Good Features to Track" J. Shi and C. Tomasi
9th IEEE Conference on Computer Vision and Pattern Recognition (1994)

Differential Techniques (Lucas-Kanade)

- Limitations:
  - The algorithm works only with small optical flow values (pixel neighborhood)
  - When there are high displacements (very close or high-speed objects) it is necessary to use a Gaussian scale pyramid.
  - When we move upwards in the pyramid (larger scale) the small movements are filtered, and the high values of optical flow become small movements.
  - In this way, we can calculate the Optical Flow at different scales



Differential Techniques (Horn-Schunck)

- Differential techniques: (Horn-Schunck)
  - Iterative method: rather than assuming that optical flow is constant in each region Q, it imposes a softness restriction that expresses the continuity of the surfaces of a rigid body.

  - Properties :
    - Iterative: convergence problems, discontinuity problems

  - Assumptions
    - The image intensity continuity equation is a good approximation of the normal component of the motion field.
    - Optical flow varies smoothly on the surfaces of a rigid body.

"Determining optical flow" B.K.P. Horn and B.G. Schunck, Artificial Intelligence, vol 17, pp 185–203, 1981

Differential Techniques (Horn-Schunck)

- Optical flow calculation: softness restriction of the Optical Flow
  - The softness restriction of the optical flow expresses the continuity of the surfaces of a rigid object.

$$\min_{\bar{v}}\left(\Gamma_C(\bar{v})\right); \quad \Gamma_C(\bar{v}) = \Gamma_S(\bar{v}) + \lambda\, \Gamma_H(\bar{v})$$

$$\bar{v} = [\, v_x(x,y,t), \;\; v_y(x,y,t)\,]^T$$

Softness Restriction

Intensity Continuity Restriction

$$\Gamma_S(\bar{v}) = \iint_{\Omega} \left[\|\nabla_x \bar{v}\|^2 + \|\nabla_y \bar{v}\|^2\right] dx\, dy$$

$$\nabla_x \bar{v} = \left[\frac{\partial v_x}{\partial x}, \frac{\partial v_y}{\partial x}\right]^T$$

$$\nabla_y \bar{v} = \left[\frac{\partial v_x}{\partial y}, \frac{\partial v_y}{\partial y}\right]^T$$

$$\Gamma_H(\bar{v}) = \iint_{\Omega} [(\nabla E)^T \bar{v} + E_t]^2\, dx\, dy$$

Ω: entorno de vecindad

"Determining optical flow" B.K.P. Horn and B.G. Schunck, Artificial Intelligence, vol 17, pp 185–203, 1981

Differential Techniques (Horn-Schunck)

- Solution for discrete images:

$$\Gamma_S(i,j) = \frac{1}{4}\left[\left(v_{x\,i+1,j} - v_{x\,i,j}\right)^2 + \left(v_{x\,i,j+1} - v_{x\,i,j}\right)^2 + \left(v_{y\,i+1,j} - v_{y\,i,j}\right)^2 + \left(v_{y\,i,j+1} - v_{y\,i,j}\right)^2 \right]$$

$$\Gamma_H(i,j) = \left[\nabla_x E(i,j,t) \cdot v_{x\,i,j} + \nabla_y E(i,j,t) \cdot v_{y\,i,j} + E_t(i,j,t) \right]^2$$

$$\boxed{\Gamma_C(i,j) = \sum_{\Omega} \left(\Gamma_S(i,j) + \lambda\, \Gamma_H(i,j)\right)}$$

Minimize $\Rightarrow$

$$\begin{cases} \dfrac{\partial\ \Gamma_C}{\partial v_{x\,i,j}}(i,j) = 0 \\ \dfrac{\partial\ \Gamma_C}{\partial v_{y\,i,j}}(i,j) = 0 \end{cases}$$

Softness Restriction

$$\frac{\partial\ \Gamma_S}{\partial v_{x\,i,j}}(i,j) = 2\left(v_{x\,i,j} - \hat{v}_{x\,i,j}\right)$$

$$\frac{\partial\ \Gamma_S}{\partial v_{y\,i,j}}(i,j) = 2\left(v_{y\,i,j} - \hat{v}_{y\,i,j}\right)$$

$\Rightarrow$ Minimize differences relative to the average value in the neighborhood

Differential Techniques (Horn-Schunck)

- Solution:

$$\Gamma_C(i,j) = \sum_{\Omega} \left(\Gamma_S(i,j) + \lambda\, \Gamma_H(i,j) \right)$$

Minimize $\Rightarrow$

$$\begin{cases} \dfrac{\partial\, \Gamma_C}{\partial v_{x\,i,j}}(i,j) = 0 \\ \dfrac{\partial\, \Gamma_C}{\partial v_{y\,i,j}}(i,j) = 0 \end{cases}$$

$$\frac{\partial\, \Gamma_C}{\partial v_{x\,i,j}}(i,j) = 2\left(v_{x\,i,j} - \hat{v}_{x\,i,j} \right) + 2\lambda \left[\nabla_x E(i,j,t) \cdot v_{x\,i,j} + \nabla_y E(i,j,t) \cdot v_{y\,i,j} + E_t(i,j,t) \right] \nabla_x E(i,j,t) = 0$$

$$\frac{\partial\, \Gamma_C}{\partial v_{y\,i,j}}(i,j) = 2\left(v_{y\,i,j} - \hat{v}_{y\,i,j} \right) + 2\lambda \left[\nabla_x E(i,j,t) \cdot v_{x\,i,j} + \nabla_y E(i,j,t) \cdot v_{y\,i,j} + E_t(i,j,t) \right] \nabla_y E(i,j,t) = 0$$

$$v_{x\,i,j} = \frac{1}{1 + \lambda\left(\nabla_x E^2 + \nabla_y E^2\right)} \cdot \left[\left(1 + \lambda \nabla_y E^2\right) \cdot \hat{v}_{x\,i,j} - \lambda \nabla_x E \left(\nabla_t E + \nabla_y E \cdot \hat{v}_{y\,i,j} \right) \right]$$

$$v_{y\,i,j} = \frac{1}{1 + \lambda\left(\nabla_x E^2 + \nabla_y E^2\right)} \cdot \left[\left(1 + \lambda \nabla_x E^2\right) \cdot \hat{v}_{y\,i,j} - \lambda \nabla_y E \left(\nabla_t E + \nabla_x E \cdot \hat{v}_{x\,i,j} \right) \right]$$

Optical flow values $\vec{v}$ in the position *(i,j)* depend on the values of $\vec{v}$ in its neighborhood

Differential Techniques (Horn-Schunck)

- Iterative solution:
  - Optical flow values $\vec{v}$ in the position *(i,j)* depend on the values of $\vec{v}$ in its neighborhood
  - We start from an initial values:

$$v^0{}_{x_{i,j}} = 0 \qquad v^0{}_{y_{i,j}} = 0 \quad \xrightarrow{\text{Iteration}}$$

$$v^{n+1}_{x_{i,j}} = v^n_{x_{i,j}} - \frac{\nabla_x E \cdot \hat{v}^n{}_{x_{i,j}} + \nabla_y E \cdot \hat{v}^n{}_{y_{i,j}} + \nabla_t E}{1 + \lambda\left(\nabla_x E^2 + \nabla_y E^2\right)} \lambda \nabla_x E$$

$$v^{n+1}_{y_{i,j}} = v^n_{y_{i,j}} - \frac{\nabla_x E \cdot \hat{v}^n{}_{x_{i,j}} + \nabla_y E \cdot \hat{v}^n{}_{y_{i,j}} + \nabla_t E}{1 + \lambda\left(\nabla_x E^2 + \nabla_y E^2\right)} \lambda \nabla_y E$$

- Problems:
  - Object discontinuities → discontinuity in optical flow

"Determining optical flow" B.K.P. Horn and B.G. Schunck, Artificial Intelligence, vol 17, pp 185–203, 1981

Example

Differential Techniques (Farneback)

- Differential techniques: (Farneback)
  - Iterative method: solves the aperture problem through polynomial interpolation in a Q-neighborhood environment, both intensity and optical flow

  - Reasons :
    - Iterative
    - Performs quadratic polynomial interpolation of the neighborhood (Q) on both frames (*polynomial expansion transform*)
    - Estimates how the polynomial surface is transformed through translation.

  - Assumptions
    - The image intensity continuity equation is imposed as the invariance of the local polynomial around each point.
    - Apparent intensity can be approximated by a displaced quadratic equation.

"Two-frame motion estimation based on polynomial expansion", Gunnar Farneback
Lecture Notes in Computer Science, 2003, (2749), 363-370.

Differential Techniques (Farneback)

- Optical flow calculation: Farneback's algorithm
  - Polynomial Expansion: polynomial approximation of the image intensity in neighborhood Q around each point:

$$f(\overline{p}) \sim \overline{p}^T \mathbf{A}\,\overline{p} + \mathbf{b}^T \overline{p} + \mathrm{c} = r_1 + r_2 x + r_3 y + r_4 x^2 + r_5 y^2 + r_6 xy$$

$$\overline{p} = [\,x,\ y]^T \quad c = r_1; \quad \mathbf{b}_{2\times1} = \begin{bmatrix} r_2 \\ r_3 \end{bmatrix}; \quad \mathbf{A}_{2\times2} = \begin{bmatrix} r_4 & {r_6}/{2} \\ {r_6}/{2} & r_4 \end{bmatrix}$$

    - Adjusting the parameters r_i by weighted minimum squares:

$$\min_{\mathbf{r}} \psi(\mathbf{r}) \ ; \quad \psi(\mathbf{r}) = \sum_{p_i \in Q} w(\overline{p}_i) \cdot \left[E(\overline{p}_i) - f(\overline{p}_i, \mathbf{r})\right]^2$$

    - Weights w_i have two components:
      - '***certainty***': intensity of neighboring pixels (image contour)
      - '***applicability***': relative position in neighborhood Q (radial)

  - Displacement estimation: Estimates how the polynomial surface is transformed to translation between two frames at the same point:

$$f_1(\overline{p}) \rightarrow f_2(\overline{p}) = f_1(\overline{p} - \overline{d}) = \overline{p}^T \mathbf{A}_2\,\overline{p} + \mathbf{b}_2^T\,\overline{p} + c_2$$

$$\mathbf{A}_2 = \mathbf{A}_1 \ ; \quad \mathbf{b}_2 = \mathbf{b}_1 - 2\mathbf{A}_1 \cdot \overline{d} \ ; \quad c_2 = \overline{d}^T \mathbf{A}_1 \overline{d} - \mathbf{b}_1^T \cdot \overline{d} + c1$$

$$\boxed{\overline{d} = -\frac{1}{2}\mathbf{A}_1^{-1}(\mathbf{b}_2 - \mathbf{b}_1)}$$

Differential Techniques (Farneback)

- Optical flow calculation: Farneback's algorithm
  - Displacement estimation : we calculate the local polynomial approximation in the Q neighborhood of each point in the image in two consecutive frames. It must be constant except for one displacement (continuity of intensity):

$$f_1(\overline{p}) = \overline{p}^T \mathbf{A}_1 \overline{p} + \mathbf{b}_1{}^T \overline{p} + c_1$$

$$f_2(\overline{p}) = \overline{p}^T \mathbf{A}_2 \overline{p} + \mathbf{b}_2{}^T \overline{p} + c_2$$

$$f_2(\overline{p}) = f_1(\overline{p} - \overline{d})$$

$$\mathbf{A}_2(\overline{p}) \approx \mathbf{A}_1(\overline{p}) \Rightarrow \mathbf{A}(\overline{p}) = \frac{\mathbf{A}_1(\overline{p}) + \mathbf{A}_2(\overline{p})}{2}$$

$$\Delta\mathbf{b}(\overline{p}) = -\frac{1}{2}(\mathbf{b}_2(\overline{p}) - \mathbf{b}_1(\overline{p}))$$

$$\mathbf{b}_2 = \mathbf{b}_1 - 2\mathbf{A}_1 \cdot \overline{d} \quad \Rightarrow \quad \boxed{\mathbf{A}(\overline{p}) \cdot \overline{\boldsymbol{d}}(\overline{p}) = \Delta\mathbf{b}(\overline{p})}$$

$$\min_{\overline{d}} \sum_{\Delta p \in Q} w(\Delta p) \cdot \left\| \mathbf{A}(\overline{p} + \Delta p) \cdot \overline{\boldsymbol{d}}(\overline{p}) - \Delta\mathbf{b}(\overline{p} + \Delta p) \right\|^2$$

$$\boxed{\overline{\boldsymbol{d}}(\overline{p}) = \left(\sum_{p_i \in Q} w_i \cdot \mathbf{A}_i^T \mathbf{A}_i \right)^{-1} \cdot \sum_{p_i \in Q} w_i \cdot \mathbf{A}_i^T \Delta\mathbf{b}_i}$$

Optical Flow

"Two-frame motion estimation based on polynomial expansion", Gunnar Farneback
Lecture Notes in Computer Science, 2003, (2749), 363-370.

Differential Techniques (Farneback)

- Optical flow calculation: Farneback's algorithm
  - Iterative algorithm:
    - Models displacement $\bar{\boldsymbol{d}}(\bar{p})$ (optical flow) as a quadratic polynomial

$$\boxed{\bar{\boldsymbol{d}}(\bar{p}) = \boldsymbol{S}(\bar{p}) \cdot \boldsymbol{a}} \qquad \boldsymbol{S}(\bar{p}) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{bmatrix} \qquad \boldsymbol{a} = [a_1\, a_2\, a_3\, a_4\, a_5\, a_6\, a_7\, a_8]^T$$

$$\boxed{\boldsymbol{a} = \left(\sum_{p_i \in Q} w_i \cdot \boldsymbol{S}_i^T \mathbf{A}_i^T \mathbf{A}_i \boldsymbol{S}_i\right)^{-1} \cdot \sum_{p_i \in Q} w_i \cdot \boldsymbol{S}_i^T \mathbf{A}_i^T \Delta \mathbf{b}_i}$$

    - Incorporates a priori knowledge: polynomial expansion is only valid locally (small displacements). The two polynomials are compared at different points from a previous prediction $\hat{\bar{p}}$
      Allows to estimate larger displacements

$$\mathbf{A}(\bar{p}) = \frac{\mathbf{A}_1(\bar{p}) + \mathbf{A}_2(\hat{\bar{p}})}{2}$$

$$\Delta \mathbf{b}(\bar{p}) = -\frac{1}{2}\left(\mathbf{b}_2(\hat{\bar{p}}) - \mathbf{b}_1(\bar{p})\right) + \mathbf{A}(\bar{p}) \cdot \hat{\bar{\boldsymbol{d}}}(\bar{p})$$

"Two-frame motion estimation based on polynomial expansion", Gunnar Farneback
Lecture Notes in Computer Science, 2003, (2749), 363-370.

Table of Contents

Estimation of the Motion Field

- Correspondence Techniques (Feature-based)
  - They estimate the motion field only at featured points.
  - These points are matched between successive images.
  - Correspondence is guided by an initial estimate of the optical flow.

  - Two methods:
    - Method of two consecutive images
    - Method of multiple consecutive images

Correspondence Techniques

- Two-image method: "*Feature Matching*" "*Lucas-Kanade*"
  - Match featured points between images in the sequence
  - Calculate A^TA over small regions of the image: Features are the centers of those regions where the smallest of the eigen values of A^TA is greater than a threshold.
  - Calculate the displacement of featured points by iterating the algorithm seen previously ⇒ estimation of the optical flow

$$A_{N^2\times 2} = \begin{bmatrix} \nabla E(\overline{p}_1) \\ \nabla E(\overline{p}_2) \\ . \\ . \\ \nabla E(\overline{p}_{NxN}) \end{bmatrix} \; ; \; b_{N^2} = \begin{bmatrix} -E_t(\overline{p}_1) \\ -E_t(\overline{p}_2) \\ . \\ . \\ -E_t(\overline{p}_{NxN}) \end{bmatrix}$$

$$\boxed{A\ \overline{v} = b}$$

$$A^T A = \begin{bmatrix} \sum(\nabla_x E)^2 & \sum \nabla_x E \cdot \nabla_y E \\ \sum \nabla_x E \cdot \nabla_y E & \sum(\nabla_y E)^2 \end{bmatrix}$$

$$\overline{v} = (A^T A)^{-1} A^T b$$

$$A^T b = \begin{bmatrix} -\sum \nabla_x E \cdot E_t \\ -\sum \nabla_y E \cdot E_t \end{bmatrix}$$

$$\overline{d} = \overline{v} \cdot \delta t$$

"Generalized Image Matching by the Method of Differences" B. D. Lucas
9th IEEE Conference on Computer Vision and Pattern Recognition (1984)

"Good Features to Track" J. Shi and C. Tomasi
9th IEEE Conference on Computer Vision and Pattern Recognition (1994)

Feature_Point_Matching Algorithm

- Input Data:
  - Two consecutive images I_1, I_2
  - A set of corresponding points in the two images
- Being:
  - Q_1, Q_2, Q' three regions of the image (*NxN*)
  - T a predefined positive real number (threshold).
  - **d** unknown displacement between I_1 e I_2 of a featured point $\overline{\boldsymbol{p}}$, centered at Q_1
- For every point $\overline{\boldsymbol{p}}$:

1.- Set **d** = 0 and center Q_1 in **p**

2.- Estimate the displacement $\mathbf{d}_0 = \overline{v} \cdot \delta t$ of $\overline{\boldsymbol{p}}$, centered at Q_1, through optical flow:

$$\bar{v} = (A^T A)^{-1} A^T b \qquad \mathbf{d} = \mathbf{d} + \mathbf{d}_0$$

3.- Q' is the region obtained from displacing Q_1 with the vector $\mathbf{d_0}$. Calculate ***S***, the sum of the square differences between Q' and the corresponding region Q_2 in I_2.

4.- If ***S*** > T, Q_1 = Q' and go back to step 1 and continue with the rest of the features; if not finish

The output is an estimation of **d** for all the featured points

Feature_Point_Matching Algorithm

- Algorithm scheme:

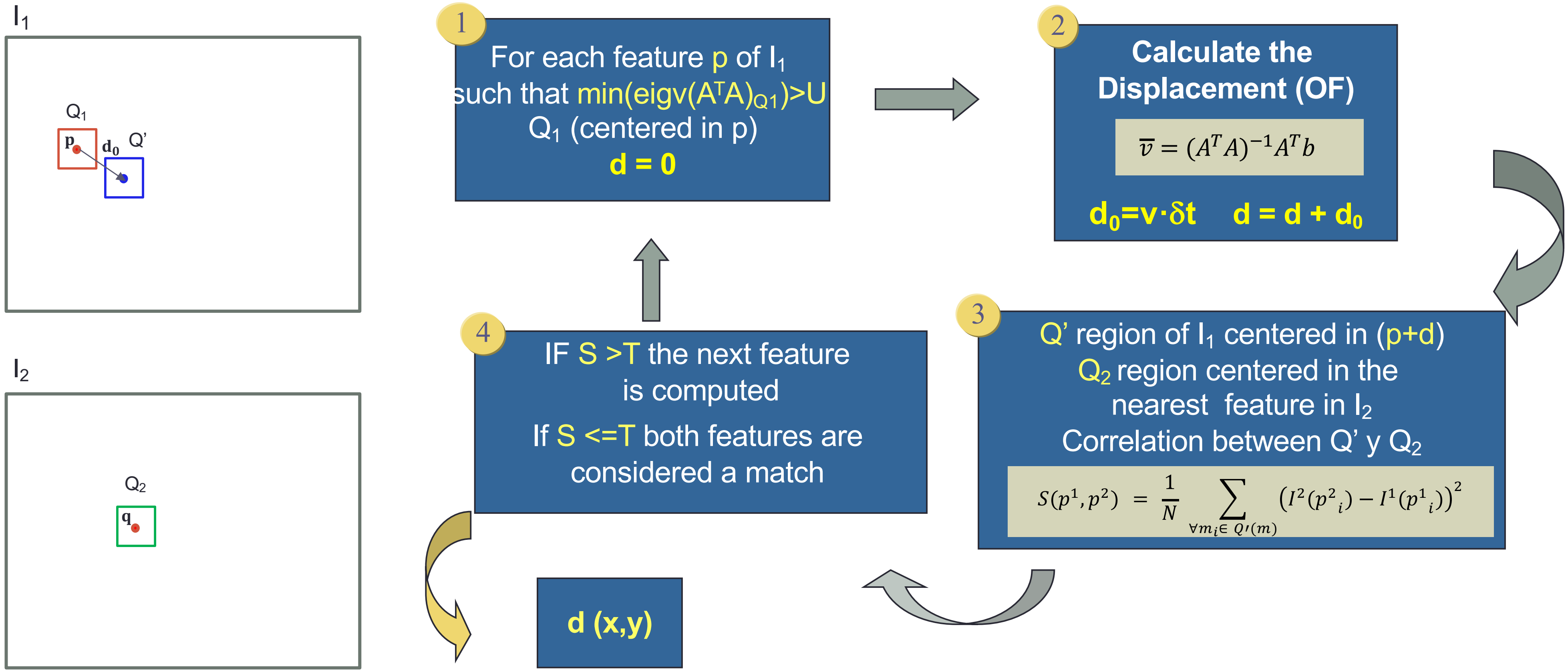


Table of Contents

Correspondence Techniques

- Multi-Frame methods
  - "*If the movement of the observed scene is continuous, as it is in most cases, predictions could be made about the movement of the points in the image, based on their previous trajectories*"
  - They use calculated disparities between I_{i-1} y I_{i-2}, I_{i-3}, I_{i-4}, .. etc. to predict the disparities between I_{i-1} y I_i before observing I_i
  - They're based on ***Kalman's Filter***..

Multiple-Frame Methods

- Kalman Filter
  - For a featured point $\overline{p}_k = [x_k , y_k]^T$ of the image in the instant t_k , with velocity $\overline{\boldsymbol{v}}_k = [v_{x,k} , v_{y,k}]^T$ $\Rightarrow$ state vector. $\mathbf{x} = [x_k , y_k , v_{x,k} , v_{y,k}]^T$
  - Kalman Filter System Model:

System/Process Model

$$\left.\begin{aligned} p_k &= p_{k-1} + v_{k-1} + \xi_{k-1} \\ v_k &= v_{k-1} + \eta_{k-1} \end{aligned}\right\} \Longrightarrow$$

System/Process Equation

$$\boxed{\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \omega_{k-1}}$$

State

$$\mathbf{x}_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix}$$

Transition Matrix

$$\Phi_{k-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

System/Process noise

$$\omega_{k-1} = \begin{bmatrix} \xi_{k-1} \\ \eta_{k-1} \end{bmatrix}$$

Measurement Equation

$$\boxed{\mathbf{z}_k = H_k\mathbf{x}_k + \mu_k}$$

Measurement noise: μ_k

$$\mathbf{z}_k = p_k + \mu_k \Longrightarrow \mathbf{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \mu_k$$

Multiple-Frame Methods

- Kalman_Tracking Algorithm
  - The inputs in t_k, are the covariance matrices of system noise and the measurement in t_{k-1}, Q_{k-1} and R_{k-1} respectively, the time-invariant state matrix Φ, time-invariant measurement matrix H, and the measurement of position in the time t_k, is z_k. The entries of P_0 are initialized with arbitrary values.

$$P'_k = \Phi_{k-1} P_{k-1} \Phi^T_{k-1} + Q_{k-1}$$

$$K_k = P'_K H_k^T \left(H_k P'_k H_k^T + R_k \right)^{-1}$$

$$\hat{\mathbf{x}}_k = \Phi_{k-1} \hat{\mathbf{x}}_{k-1} + K_k (\mathbf{z}_k - H_k \Phi_{k-1} \hat{\mathbf{x}}_{k-1})$$

$$P_k = (I - K_k H_k) P'_k$$

Covariance Matrices:

Q_{k-1}: system noise covariance.
R_{k-1}: measurement noise covariance.
P'_k : a priori error covariance.
P_k : a posteriori error estimation covariance

K_k: Kalman Filter Gain matrix

  - Output is the optimal estimation of position and velocity in the time t_k, $\hat{\mathbf{x}}_k$ and its uncertainties, are given by the elements of the diagonal of P_k

Multiple-Frame Methods

- Conclusions :
  - Integrates noise measurements with model predictions to estimate optimal state
  - The filter quantifies the uncertainty of the estimated state, as the diagonal elements of the state covariance matrix. This allows the feature detector to automatically resize the region of the image to be searched to find the featured point in the next image. The region is focused on the best estimated position and a region as large as the uncertainty.

Multiple-Frame Methods

- Problems implementing this algorithm:
  - Lost Information: featured points may not be found in some frames
  - Kalman's filter is based on knowledge of the following data:
    - The model of the system and its corresponding covariance matrix Q_k
    - The measured model and the corresponding noise covariance matrix, R_k
    - The initial system state in t_0, $\hat{x}_0$, and the state covariance matrix, P_0

    However many of these terms are unknown.
  - An independent Kalman filter is required for each featured point

Table of Contents

3D Reconstruction

- Assuming the motion field (projected velocity maps on the image) is known, what can we know about the 3D movement and location of the objects in the scene?
- We know the displacements of each point (or certain features) of the image between two successive images.

$$\boxed{\overline{d} = \overline{v} \cdot \delta t}$$

- Without any additional restrictions the problem of 3D reconstruction has no a unique solution, therefore we must study particular situations:
  - Static scene and mobile camera rotating around its projection center (unknown rotation)
  - Static scene and mobile camera with unknown movement
  - Calibrated stationary camera and mobile object with known motion

3D Reconstruction

- Static scene and mobile camera rotating around its projection center (unknown rotation)
  - Given a set of corresponding points between two images I_1 and I_2 taken in instants t_1 and t_2
  - We can calculate the rotation made by the camera if we have at least 3 corresponding points
  - There is a Homography between the two views $\boxed{\tilde{m}^2 \cong H_\theta \cdot \tilde{m}^1}$
  - We cannot infer anything about the 3D position of the projected points and therefore the surface of the objects

$$\left.\begin{aligned} x^{(1)} &= \frac{X^{C1}}{Z^{C1}} \\ y^{(1)} &= \frac{Y^{C1}}{Z^{C1}} \end{aligned}\right\} \qquad \left.\begin{aligned} x^{(2)} &= \frac{r_{11}X^{C1} + r_{12}Y^{C1} + r_{13}Z^{C1}}{r_{31}X^{C1} + r_{32}Y^{C1} + r_{33}Z^{C1}} \\ y^{(2)} &= \frac{r_{21}X^{C1} + r_{22}Y^{C1} + r_{23}Z^{C1}}{r_{31}X^{C1} + r_{32}Y^{C1} + r_{33}Z^{C1}} \end{aligned}\right\} \Rightarrow \left.\begin{aligned} x^{(2)} &= \frac{r_{11}x^{(1)} + r_{12}y^{(1)} + r_{13}}{r_{31}x^{(1)} + r_{32}y^{(1)} + r_{33}} \\ y^{(2)} &= \frac{r_{21}x^{(1)} + r_{22}y^{(1)} + r_{23}}{r_{31}x^{(1)} + r_{32}y^{(1)} + r_{33}} \end{aligned}\right\}$$

Normalized Coordinates

$$\tilde{p}^{(1)} = A^{-1}\tilde{m} = [I \quad 0]\ \tilde{M}_{c1}$$

$$\tilde{p}^{(2)} = A^{-1}\tilde{m} = [R \quad t]\ \tilde{M}_{c1}$$

$$\det|R| = 1$$

$$R^T = R^{-1}$$

$$\boxed{[R]} \Leftarrow$$

$$\boxed{\tilde{H}_\theta \cong A\,RA^{-1}}$$

3D Reconstruction

- Static scene and mobile camera with unknown movement
  - Given a set of corresponding points between two images I_1 and I_2 taken in instants t_1 and t_2
  - We can calculate the rotation and translation performed and the position of the 3D points if we have at least 11 corresponding points

$$\left.\begin{aligned} x^{(2)} &= \frac{r_{11}X^{C1} + r_{12}Y^{C1} + r_{13}Z^{C1} + t_1}{r_{31}X^{C1} + r_{32}Y^{C1} + r_{33}Z^{C1} + t_3} \\ y^{(2)} &= \frac{r_{21}X^{C1} + r_{22}Y^{C1} + r_{23}Z^{C1} + t_2}{r_{31}X^{C1} + r_{32}Y^{C1} + r_{33}Z^{C1} + t_3} \end{aligned}\right\} \Rightarrow \left.\begin{aligned} x^{(2)} &= \frac{r_{11}x^{(1)} + r_{12}y^{(1)} + r_{13} + \frac{t_1}{Z^{c1}}}{r_{31}x^{(1)} + r_{32}y^{(1)} + r_{33} + \frac{t_3}{Z^{c1}}} \\ y^{(2)} &= \frac{r_{21}x^{(1)} + r_{22}y^{(1)} + r_{23} + \frac{t_2}{Z^{c1}}}{r_{31}x^{(1)} + r_{32}y^{(1)} + r_{33} + \frac{t_3}{Z^{c1}}} \end{aligned}\right\}$$

$$\left.\begin{aligned} x^{(1)} &= \frac{X^{C1}}{Z^{C1}} \\ y^{(1)} &= \frac{Y^{C1}}{Z^{C1}} \end{aligned}\right\}$$

$$\phi_i\left([R], t, Z_i, \left\langle x_i^{(1)}, y_i^{(1)}, x_i^{(2)} \right\rangle\right) = 0$$

$$\Psi_i\left([R], t, Z_i, \left\langle x_i^{(1)}, y_i^{(1)}, y_i^{(2)} \right\rangle\right) = 0$$

- Same problem as the visual odometry seen in the previous topic. Optical flow allows us to have dense map (upto a scale)

$$\boxed{\tilde{q}^{(k)^T} \; E \; \tilde{q}^{(1)} = 0}$$

$$\boxed{E \cong \left[{}^{(k)}t_{(k+1)}\right]_{[\wedge]} \cdot {}^{(k)}R_{(k+1)}}$$

Reconstrucción 3D

- Calibrated stationary camera and mobile object with known motion

$$\bar{M}^t = [X^t \quad Y^t \quad Z^t]^T$$

Coordinates of the 3D point referred to the C.S. of camera at the instant t

$$\bar{p}^{\,t} = \frac{\bar{M}^t}{Z^t}$$

$$\bar{p} = [x\,,\ y,\ 1]^{\,T}$$

Normalized coordinates of the projected point on the image referred to the C.S.. of camera (3D) at the instant t

$$\tilde{p} = A^{-1}\tilde{m} \cong \ [I \quad 0]\ \tilde{M}_c$$

$$\mathbf{z}^{\,t} = \mathbf{H} \cdot \bar{p}^{\,t} + \mu_k$$

Measurement of 2D coordinates projected into the image (with noise).

$$\boldsymbol{R}_\delta\ ,\ \boldsymbol{T}_\delta$$

Rotation and translation between t and t+1 instants (known)

$$\bar{M}^{t+1} = \boldsymbol{R}_\delta \cdot \bar{M}^t + \boldsymbol{T}_\delta$$

Reconstrucción 3D

- Clearing M^{t+1} and M^t in the above equations we have:

$$\boldsymbol{R}_{\delta}{}^{T}\overline{p}^{\,t+1} \cdot Z^{t+1} - \overline{p}^{\,t} \cdot Z^{t} = \boldsymbol{R}_{\delta}{}^{T}\boldsymbol{T}_{\delta}$$

$with\ \ \overline{p}_i$: same depth points (neighborhood)

$$\begin{bmatrix} \boldsymbol{R}_{\delta}{}^{T}\overline{p}_1^{\,t+1} & -\overline{p}_1^{\,t} \cdot Z^{t} \\ \vdots & \vdots \\ \boldsymbol{R}_{\delta}{}^{T}\overline{p}_n^{\,t+1} & -\overline{p}_n^{\,t} \cdot Z^{t} \end{bmatrix} \cdot \begin{bmatrix} Z^{t} \\ Z^{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\delta}{}^{T}\boldsymbol{T}_{\delta} \\ \vdots \\ \boldsymbol{R}_{\delta}{}^{T}\boldsymbol{T}_{\delta} \end{bmatrix}$$

$$\boxed{A \cdot \bar{Z} = b}$$

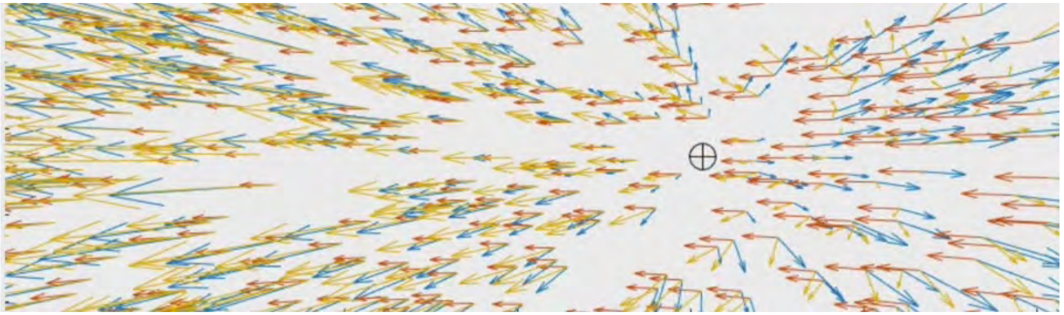
$\Downarrow$

3D point Coordinates referred to the camera SC.S. at the instant t

$$\bar{Z} = \begin{bmatrix} Z^{t} \\ Z^{t+1} \end{bmatrix} = (A^{T}A)^{-1}A^{T}b \quad \Rightarrow \quad \boxed{\bar{M}^{t} = Z^{t} \cdot \overline{p}^{\,t}}$$

The Motion Field and 3D Reconstruction

- Motion Field decomposition



$$v_x = \frac{T_z x - T_x}{Z} - \omega_y + \omega_z y + \omega_x xy - \omega_y x^2$$
$$v_y = \frac{T_z y - T_y}{Z} + \omega_x - \omega_z x - \omega_y xy + \omega_x y^2$$

$$\overline{v} = \frac{1}{Z}\mathbf{F}_{2\times3}(x,y)\cdot\overline{T} + \mathbf{G}_{2\times3}(x,y)\cdot\overline{\omega}$$

$$\overline{v} = \left[\mathbf{F}_{2\times3}(x,y)\cdot\overline{T} \quad \mathbf{G}_{2\times3}(x,y)\right]\cdot\begin{bmatrix}{}^{1}/_{Z}\\ \overline{\omega}\end{bmatrix}$$

$$Unknowns \rightarrow \left[\overline{T}, \quad \overline{\omega}, \quad Z_1, \cdots, Z_n\right]$$

$$\underset{\overline{T}\in S^2}{arg\ \boldsymbol{min}} \left\| \overline{v} - \Phi(\overline{T})\cdot\Phi(\overline{T})^{+}\overline{v} \right\|$$

From a set of points ($\boldsymbol{n}$) we can get a system of linear equations:

$$\begin{bmatrix}\overline{v}_1\\ \overline{v}_2\\ \vdots\\ \overline{v}_n\end{bmatrix} = \Phi_{n\times(2n+3)}\left(\overline{T}\right)\cdot\begin{bmatrix}{}^{1}/_{Z_1}\\ {}^{1}/_{Z_2}\\ \vdots\\ {}^{1}/_{Z_n}\\ \overline{\omega}\end{bmatrix}$$

1) Get the translation vector (velocity)
2) From the previous system we calculated the rotational velocity and depth

Table of Contents

……Fin